

WORKMAN NYDEGGER
A PROFESSIONAL CORPORATION
ATTORNEYS AT LAW
1000 EAGLE GATE TOWER
60 EAST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84111
TELEPHONE (801) 533-9800
FAX (801) 328-1707

TELECOPIER COVER SHEET

March 2, 2010

Total Number of Pages
(including cover letter): 11 Pages

Please deliver the transmitted facsimile pages to:

EXAMINER Isaac Tuku Tecklu
GROUP ART UNIT: 2192

Telecopier Phone: (571) 273-7957

From: Brian Tucker/Chantell Johnston

Serial No.: 10/824,969

Interview: Wednesday, March 10, 2010 at 1:30 PM in the Randolph Building

Docket No.: 13768.1294

Customer No.: 47973

We are transmitting from a Sharp FO-750 or Sharp FO-6100 facsimile machine. If you do not receive all the pages or they are unreadable, please contact me as soon as possible at (801) 533-9800.

THE INFORMATION CONTAINED IN THIS FACSIMILE MESSAGE IS ATTORNEY PRIVILEGED AND CONFIDENTIAL INFORMATION INTENDED ONLY FOR THE USE OF THE INDIVIDUAL OR ENTITY NAMED ABOVE. IF THE READER OF THIS MESSAGE IS NOT THE INTENDED RECIPIENT, OR THE EMPLOYEE OR AGENT RESPONSIBLE TO DELIVER IT TO THE INTENDED RECIPIENT, YOU ARE HEREBY NOTIFIED THAT ANY DISSEMINATION, DISTRIBUTION OR COPYING OF THIS COMMUNICATION IS STRICTLY PROHIBITED. IF YOU HAVE RECEIVED THIS COMMUNICATION IN ERROR, PLEASE IMMEDIATELY NOTIFY US BY TELEPHONE, AND RETURN THE ORIGINAL MESSAGE TO US AT THE ABOVE ADDRESS VIA THE U.S. POSTAL SERVICE. THANK YOU.

BRIAN TUCKER

OFFICE DIRECT: 801-322-8427

EMAIL: btucker@WNLAW.com

VIA eFILE

PATENT APPLICATION
Docket No. 13768.1294

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

| | | |
|----------------------|-----------------------------|------------|
| In re application of | Brian D. Harry, et al. |) |
| | |) |
| Serial No.: | 10/824,969 |) Art Unit |
| | |) 2192 |
| Filed: | April 15, 2004 |) |
| | |) |
| Conf. No.: | 7457 |) |
| | |) |
| For: | OFFLINE SOURCE CODE CONTROL |) |
| | |) |
| Examiner: | Isaac Tuku Tecklu |) |
| | |) |
| Customer No.: | 47973 |) |

AMENDMENT "G" AND RESPONSE AFTER NON-FINAL

ELECTRONICALLY FILED
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

In response to the Office action mailed February 18, 2010 (paper no. 20100203), please amend the above-identified application as follows:

Amendments to the Claims are reflected in the listing of claims which begins on page 2 of this paper;

Remarks/Arguments begin on page «Form Data» of this paper.

Application No. 10/824,969
Amendment "G" dated ~~8/1/2010~~
Reply to Non-Final Office Action mailed February 18, 2010

AMENDMENTS TO THE CLAIMS

This listing of claims replaces all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended) A client system that facilitates source code control, comprising:

a processor; and

one or more physical computer readable storage media operatively coupled to the processor, the computer-readable storage media having stored thereon computer executable instructions that, when executed by the processor, are configured to implement the client system, including:

a client-side client workspace that stores at least one source code file downloaded from a server-side source control repository that maintains state information for one or more source code files stored in the repository, the client workspace enabling a user of the client system to checkout the source code file for modification by the user and to store the source code file when modified;

a client-side client source code control component that tracks an activity associated with a modification of the source code file in the client workspace when the client is in an offline mode, and that transmits the activity during an update process when the client moves to an online mode, the client source code control component comprising:

a file cache that stores a pristine copy of the source code file in an unmodified state in response to the source code file being checked out for modification from the client-side client workspace; and

an activity list that stores the activity, the activity comprising one or more commands executed against the source code file during the offline mode and associated with the modification; and

wherein the client-side client source code control component is configured to interface with a server-side server source code control component to facilitate transfer of the activity and update of the source code file to the server-side source control repository, ~~wherein one of the client-side client source code control component or the server-side server source code control component checks for an error during the update~~

Application No. 10/824,969

Amendment "G" dated ~~8/1/2010~~ Form Data

Reply to Non-Final Office Action mailed February 18, 2010

~~process and determines whether the update can proceed or must be aborted in part or whole.~~

2. (Canceled)

3. (Previously Presented) The client system of claim 1, wherein the source code file stored in the file cache remains in an unmodified state.

4. (Previously Presented) The client system of claim 1, wherein contents of the file cache are maintained in both the offline mode and online mode of the client.

5. (Previously Presented) The client system of claim 1, wherein the client workspace stores all source code files that have been at least one of modified and deleted.

6. (Previously Presented) The client system of claim 1, wherein the client downloads the source code file from the server-side source control repository before the client moves to the offline mode.

7. (Canceled)

8. (Previously Presented) The client system of claim 1, wherein the file cache also stores at least one of pending changeset data, a filetype definition, and a site-specific help file.

9. (Previously Presented) The client system of claim 1, wherein an error is resolved during a reconciliation process of the activity to the source code file before the source code file can be updated with the modification.

10. (Canceled)

11. (Previously Presented) The client system of claim 1, wherein the source code file is downloaded into the client workspace before the client moves to the offline mode.

Application No. 10/824,969

Amendment "G" dated ~~4/1/2010~~

Reply to Non-Final Office Action mailed February 18, 2010

12. (Previously Presented) The client system of claim 1, wherein a pristine copy of the source code file is automatically loaded into a client file cache in response to a checkout-related command being executed.

13. (Previously Presented) The client system of claim 1, wherein the activity list stores the activity in an arbitrary order which need not be sequential, such that the client checks the activity list for an error during the update process and determines whether to transmit the activity list when the client moves to the online mode or to abort a part or all of the update process.

14. (Previously Presented) The client system of claim 1, wherein the source code file stored in the file cache remains in an unmodified state, and is a file that has been at least one of modified and deleted in the client workspace.

15. (Previously Presented) The client system of claim 1, wherein the source code file is stored in the file cache of the client during a checkout process that is executed when the client is in one of the offline mode and the online mode.

16. (Canceled)

17. (Previously Presented) The client system of claim 1, wherein the activity is persisted to a server to update a server source code file associated with the source code file during an update process associated with the online mode.

18-19. (Canceled)

20. (Previously Presented) The client system of claim 1, further comprising a classifier that automates a source code control feature by making an inference based on data associated with at least one of the online mode and the offline mode.

Application No. 10/824,969

Amendment "G" dated ~~6/2/2010~~

Reply to Non-Final Office Action mailed February 18, 2010

21. (Previously Presented) In a network comprising a client system that facilitates course code control, a computer program product for implementing a method of tracking and reconciling offline file editing and enforcing file security, the computer program product comprising one or more physical computer-readable storage media having stored thereon computer-executable instructions that, when executed by a processor, cause the client system to perform the following::

- establishing a client workspace to store a plurality of source code files, the client workspace established to allow a user of the client to checkout a source code file for modification by the user and to store the source code file when modified;

- downloading a copy of a remote source code file from a server source code control repository to the client workspace;

- moving the client system to an offline mode;

- executing one or more commands to checkout the source code file in the client workspace for editing by the user, and in response to the checkout commands, caching a pristine copy of the source code file in a client file cache, wherein the pristine copy is usable in the offline mode to facilitate undo and difference processes;

- modifying the source code file in the client workspace by executing one or more commands against the source code file;

- storing activity data in an activity list, which activity data includes the one or more commands executed against the source code file;

- moving the client system to an online mode;

- performing an error check to determine if a security error exists, including:

- determining if the remote source code file is locked; and

- determining if an administrator has disallowed an update process;

- reconciling the activity data with the remote source code file by transmitting the activity data to the server to update the remote source code file if no security error is detected;

- resolving any conflicts that occur during the reconciliation process; and

- uploading the modified source code file to the server source code control repository when any conflicts have been resolved.

Application No. 10/824,969

Amendment "G" dated 3/2/2010

Reply to Non-Final Office Action mailed February 18, 2010

22-27. (Canceled)

28. (Previously Presented) At a client system that facilitates source code control, a method of tracking and reconciling offline file editing and enforcing file security, the method comprising acts of:

- establishing a client workspace to store a plurality of source code files, the client workspace established to allow a user of the client to checkout a source code file for modification by the user and to store the source code file when modified;

- downloading a copy of a remote source code file from a server source code control repository to the client workspace;

- moving the client system to an offline mode;

- executing one or more commands to checkout the source code file in the client workspace for editing by the user, and in response to the checkout commands, caching a pristine copy of the source code file in a client file cache, wherein the pristine copy is usable in the offline mode to facilitate undo and difference processes;

- modifying the source code file in the client workspace by executing one or more commands against the source code file;

- storing activity data in an activity list, which activity data includes the one or more commands executed against the source code file;

- moving the client system to an online mode;

- performing an error check to determine if a security error exists, including:

 - determining if the remote source code file is locked; and

 - determining if an administrator has disallowed an update process;

- reconciling the activity data with the remote source code file by transmitting the activity data to the server to update the remote source code file if no security error is detected;

- resolving any conflicts that occur during the reconciliation process; and

- uploading the modified source code file to the server source code control repository when any conflicts have been resolved.

Application No. 10/824,969

Amendment "G" dated ~~9/1/2010~~

Reply to Non-Final Office Action mailed February 18, 2010

29. (Original) The method of claim 28, further comprising storing information at the client before entering the offline mode.

30. (Original) The method of claim 28, further comprising updating a checkout record at the server during the online mode.

31. (Original) The method of claim 28, further comprising issuing a command to enter the offline mode, and a corresponding command to enter the online mode.

32. (Original) The method of claim 28, further comprising issuing a command that includes a URL to a workspace.

33. (Previously Presented) The method of claim 28, further comprising: checking out a source code file during the act of downloading, which is during the online mode.

34. (Original) The method of claim 28, further comprising: detecting an error during the offline mode; presenting an error message associated with the error; maintaining the client in the offline mode in response to detecting the error; and allowing the client to move to the online mode after the error has been resolved.

35. (Original) The method of claim 28, further comprising imposing permissions required for the offline mode, during the online mode.

36. (Original) The method of claim 28, further comprising caching at the client at least one of unmodified files, pending change set information, filetype definitions, and site-specific help files.

37. (Previously Presented) The method of claim 28, further comprising reapplying a checkout process to the server when at least one of a checkout was cancelled at the server when

Application No. 10/824,969

Amendment "G" dated ~~10/11/2010~~

Reply to Non-Final Office Action mailed February 18, 2010

the client was offline and a checkout was performed offline on the client after the source code file was downloaded to the client without the checkout process issued to the server during the online mode.

38-41. (Canceled)

DRAFT

Application No. 10/824,969
Amendment "G" dated ~~8/24/2009~~
Reply to Non-Final Office Action mailed February 18, 2010

REMARKS

The Office Action mailed February 18, 2010, considered and rejected claims 1, 3-6, 8, 9, 11-15, 17, 20, 21 and 28-37. Claims 1, 3-6, 8, 9, 11-15, 17, 20, 21 and 28-37 were rejected under 35 U.S.C. § 102(e) as being anticipated by *Lin et al.* (U.S. Publ. No. 2005/0091226).

Traversal of the Anticipation Rejections

The present invention is directed to a source code control ("SCC") application that provides enhanced functionality while working in the offline state. For example, the present invention, as addressed in the previous response, facilitates undo and difference operations by caching a pristine copy of a source code file when it is checked out for modification. The modifications are not made to this pristine copy, but to the actual checked out source code file. It is noted that there are two different files involved in this process: (1) the source code file itself that was downloaded to the client-side client workspace, and (2) the pristine copy of the source code file that is created when the source code file is checked out from the client-side client workspace for modification. Both of these files are on the client and are different from the actual source code file on the server-side source control repository.

The examiner has again maintained his 102(e) rejections in the current action. Applicant submits, however, that in order for a reference to anticipate a claim, each limitation must be taught by the reference. This is not the case with *Lin*. *Lin* is not directed to a SCC system. Source code is not even mentioned in *Lin*. To the contrary, *Lin* is directed to SMB and DFS. Even if the aspects of how *Lin* and the present invention function were similar, *Lin* could only anticipate the present claims if it teaches every claimed aspect. The claims contain specific limitations that clearly indicate that they are directed to a source code control system. The examiner is reading these limitations out of the claims in order to retain the 102(e) rejection.

In the Response to Arguments section of the Office Action, the examiner argues that the caching of the files to the local data store in *Lin* is the same as the present invention. This cannot be so. In *Lin*, there is a single cached copy of the file. This cached copy is modified directly. In contrast, the present invention downloads (or caches) the source code file first in the client-side client workspace. Then, when it is checked out, a separate pristine copy of the source code file is created and stored in the file cache of the client-side client source code control component. At this point, there are two copies of the source code file on the client. *Lin* discloses nothing similar to this, and therefore cannot teach these limitations. Again, it is emphasized that for a reference

Application No. 10/824,969
Amendment "G" dated ~~1/1/2010~~
Reply to Non-Final Office Action mailed February 18, 2010

to anticipate, it must teach every limitation. Because Lin does not teach these limitations, it cannot anticipate the independent claims. Applicant therefore requests that the current rejections be withdrawn.

In view of the foregoing, Applicant respectfully submits that the other rejections to the claims are now moot and do not, therefore, need to be addressed individually at this time. In the event that the Examiner finds remaining impediment to a prompt allowance of this application that may be clarified through a telephone interview, the Examiner is requested to contact the undersigned attorney at (801) 322-8427.

Dated this ____ day of ____, 2010.

Respectfully submitted,

RICK D. NYDEGGER
Registration No. 28,651
BRIAN D. TUCKER
Registration No. 61,550
Attorneys for Applicant
Customer No. 47973

RDN:BDT:cj
2756616_1